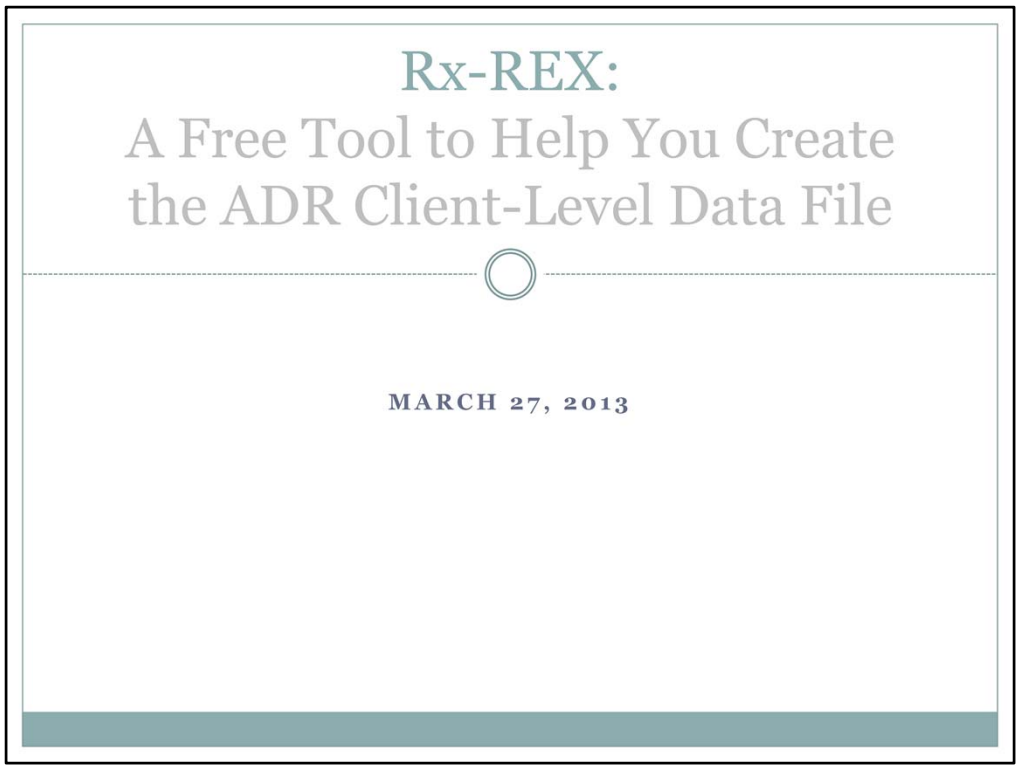
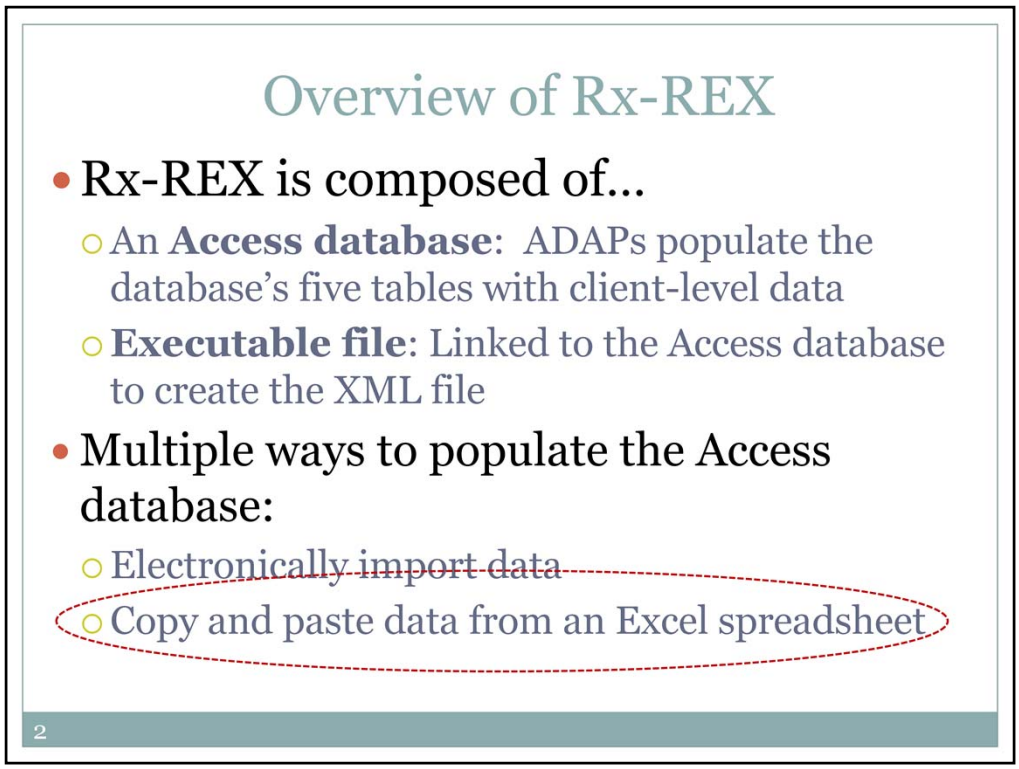


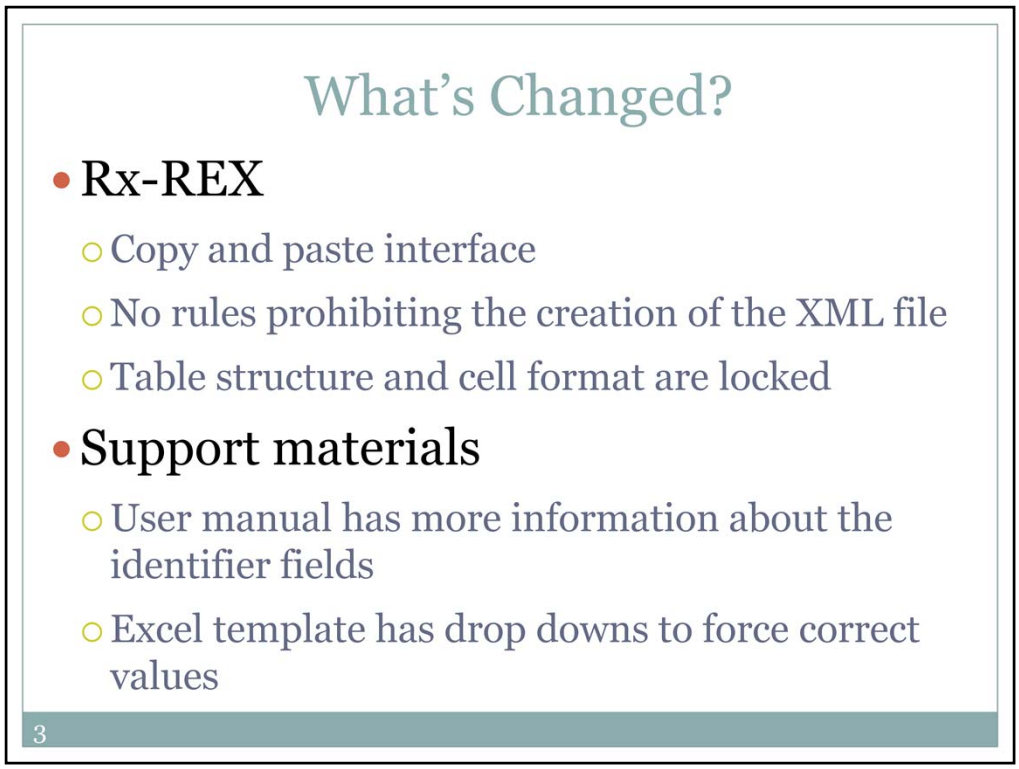
HAB and its contractor, SAIC, developed Rx-REX to help ADAPs create the ADR client-level data file. During this webcast, we'll walk you through the process of using Rx-REX.


Overview of Rx-REX
Rx-REX is composed of…
An Access database: ADAPs populate the database's five tables with client-level data
Executable file: Linked to the Access database to create the XML file
Multiple ways to populate the Access database:
Electronically import data
Copy and paste data from an Excel spreadsheet
2

Let's start with an overview of Rx-REX. Rx-REX is composed of two main applications: there is an Access database that you populate with your client-level data. And, there is an executable file that is linked to the Access database to create the client-level data XML file.

There are multiple ways to get your data into the Access database. You can develop your own programming code or use a mapping software to electronically import data from your data management system into the Access database.

Or, you can copy and paste data from spreadsheets. This strategy is the most basic, requiring the least technical resources. This is the strategy we'll describe today and that is outlined in the user manual.


What's Changed?
Rx-REX
Copy and paste interface
No rules prohibiting the creation of the XML file
Table structure and cell format are locked
Support materials
User manual has more information about the identifier fields
Excel template has drop downs to force correct values
3

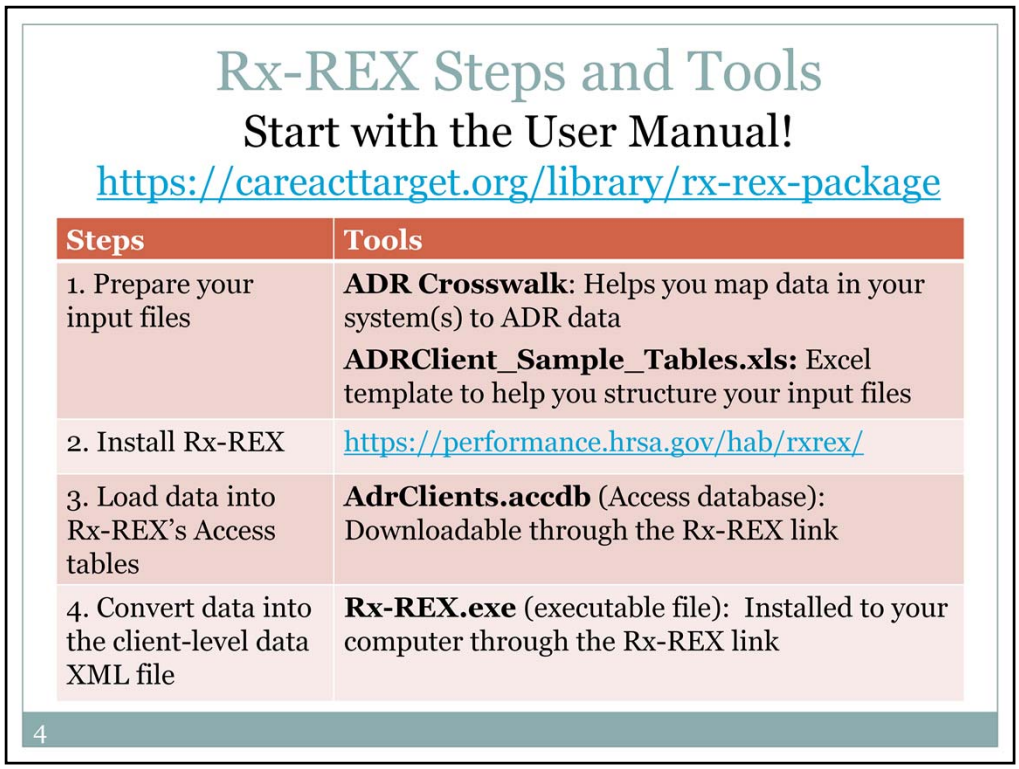
Many of you participated in the Rx-REX beta test. Thank you! We really appreciate the time you dedicated to the process and your feedback. SAIC made several changes to the tool based on your experience.

First, they developed a new copy and paste interface to facilitate that process.

They also removed all of the data validation rules that prohibited the creation of the XML file. This is because HAB just wants ADAPs to focus on creating a compliant XML file for this submission. For future submissions, we'll start looking at missing data and inconsistent data values.

They also locked the table structure so ADAPs could not force their data into Rx-REX. You must input your data into Rx-REX using the ADR-required values.

Based on your feedback, we also updated the support material. We added more information about the identifier fields. We also incorporated drop down menus into the Excel template, once again to help you structure your data with the correct values.


Rx-REX Steps and Tools
Start with the User Manual!
https://careacttarget.org/library/rx-rex-package
Steps
Tools
1. Prepare your input files
ADR Crosswalk: Helps you map data in your system(s) to ADR data
ADRClient_Sample_Tables.xls: Excel template to help you structure your input files
2. Install Rx-REX
https://performance.hrsa.gov/hab/rxrex/
3. Load data into Rx-REX's Access tables
AdrClients.accdb (Access database): Downloadable through the Rx-REX link
4. Convert data into the client-level data XML file
Rx-REX.exe (executable file): Installed to your computer through the Rx-REX link
4

Rx-REX requires four mains steps: we'll go over these and the tools associated with each step now.

All of these steps are described in detail in the Rx-REX user manual. That should be the first place to go if you have questions about the application.

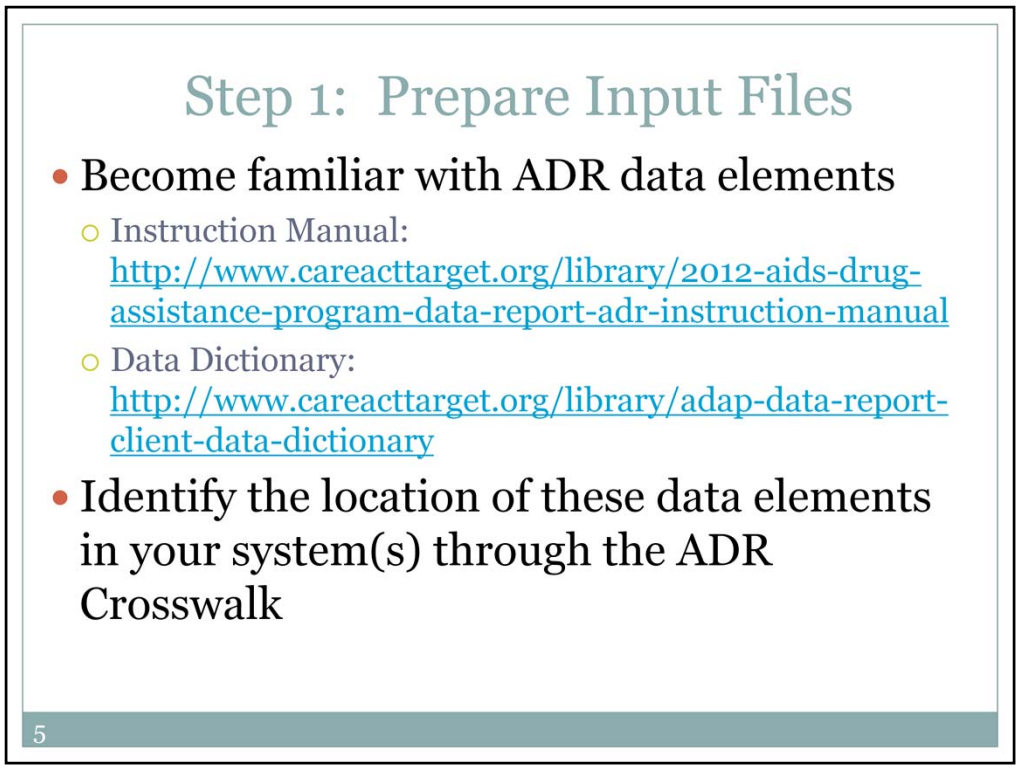
Step one: You need to prepare your input files. This is by far the hardest and most time consuming step. To help with this process, we have developed two tools. An ADR Crosswalk, which allows you to map on paper the data elements in your system to the data elements in the ADR. We have also developed an Excel template that can help you structure your input files to mirror Rx-REX five Access tables. These tools are located in a zip package along with the user manual on TARGET. The link is at the top of the slide.

Second, you install Rx-REX through a link. You can find this link on the slide and on TARGET.

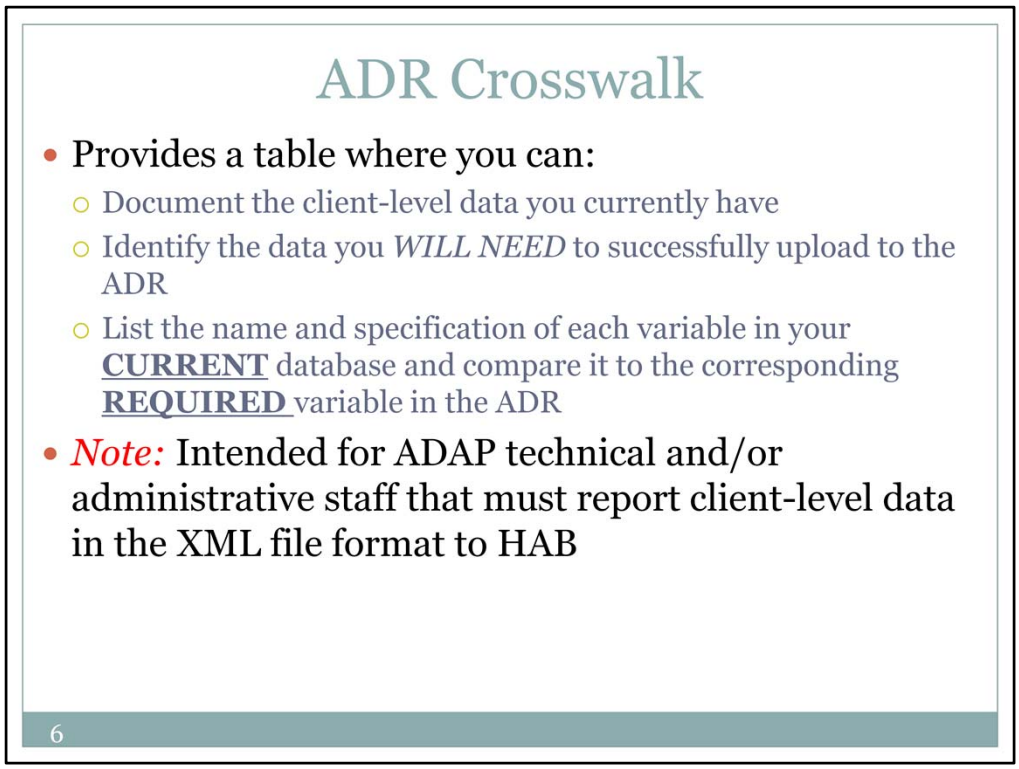
Third, you load your data into Rx-REX's Access database, called AdrClients.

Fourth, you use the executable file to create the client-level data file.

Now, let's go over these steps in more detail.

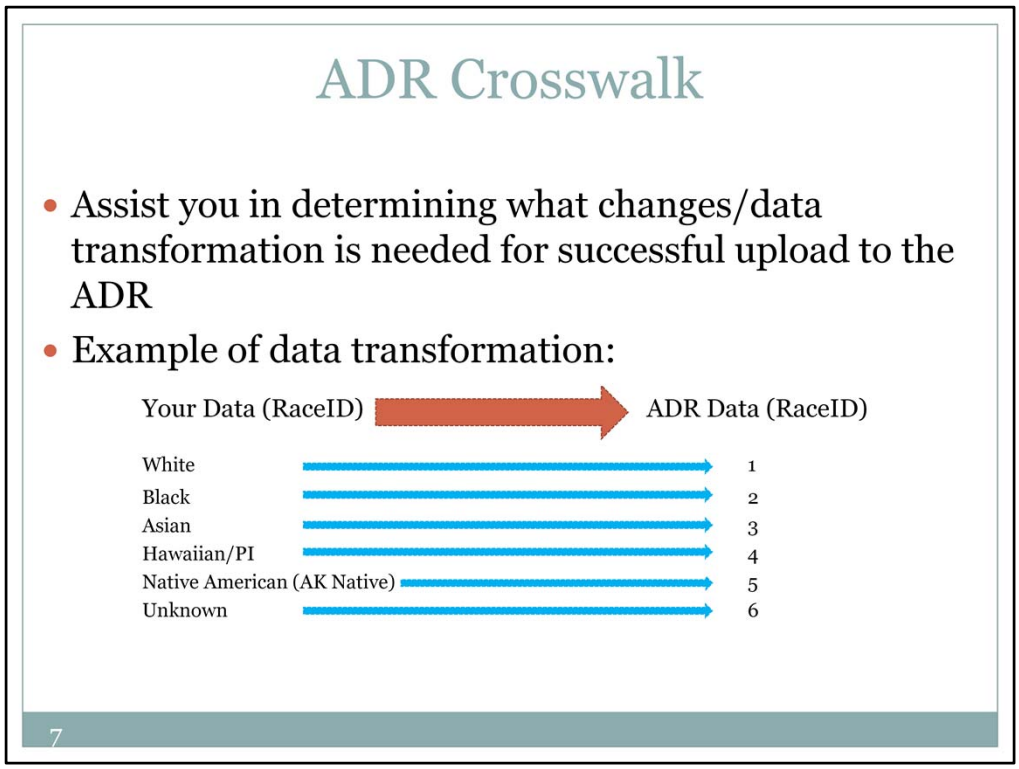

Step 1: Prepare Input Files
Become familiar with ADR data elements
Instruction Manual: http://www.careacttarget.org/library/2012-aids-drug-assistance-program-data-report-adr-instruction-manual
Data Dictionary: http://www.careacttarget.org/library/adap-data-report-client-data-dictionary
Identify the location of these data elements in your system(s) through the ADR Crosswalk
5

To start preparing your input files, you need to become very familiar with the ADR data elements. If you haven't already done so, check out the instruction manual and the data dictionary.


ADR Crosswalk
Provides a table where you can:
Document the client-level data you currently have
Identify the data you WILL NEED to successfully upload to the ADR
List the name and specification of each variable in your CURRENT database and compare it to the corresponding REQUIRED variable in the ADR
Note: Intended for ADAP technical and/or administrative staff that must report client-level data in the XML file format to HAB
6

The "crosswalk" is intended to help you identify the data you currently have and the data you will need to successfully upload to the ADAP Data Report (ADR). The "crosswalk" provides a table in which you can list the name and specification of each variable in your current database and compare it to the name and specification of the corresponding variable in the ADR.

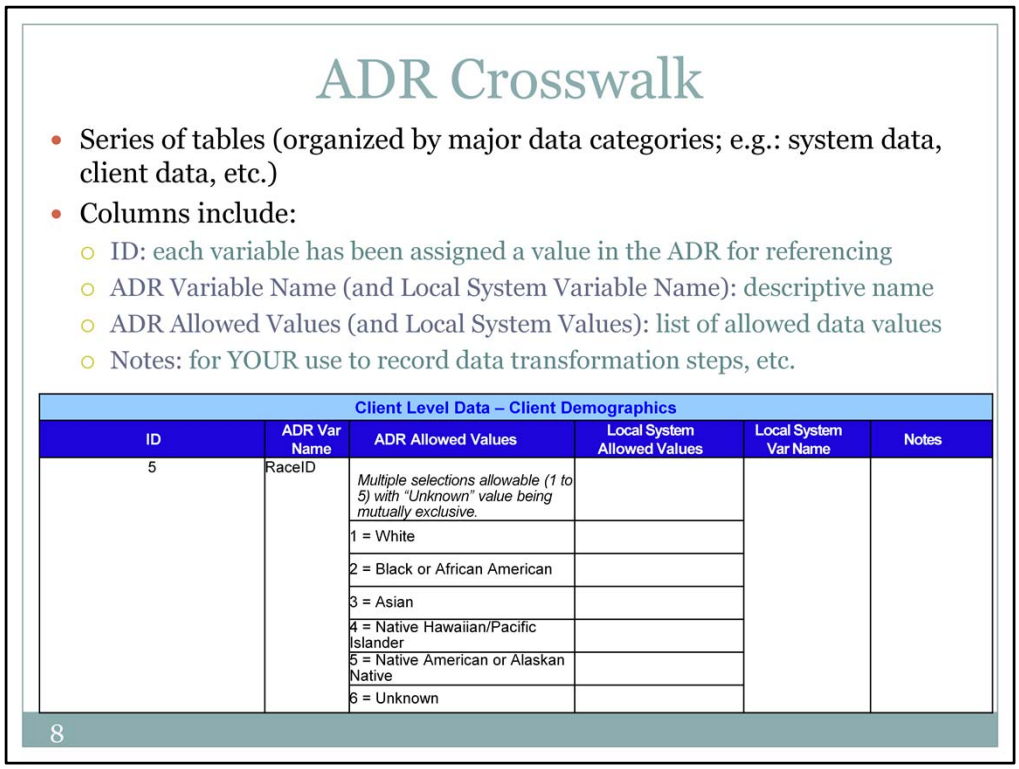
Please note, that this "crosswalk" is only intended for ADAP technical and/or administrative staff that are involved with reporting client-level data in the XML file format to HAB



This crosswalk will also help you assess the data you have and determine what, if any changes are necessary to correctly format and report your data using the ADR.

You will use the Crosswalk both as a guide for determining how you need to modify your data from its current format into a format accepted by the ADR (call data transformation), as well as a record of what you did.

In this example, with the RaceID variable, your own database might categorize clients as “white”, “black”, etc. However, the ADR requires this variable be categorized as numbers 1 through 6. Using this crosswalk, it should be easier to see and track what data transformations are necessary, as “white” becomes “1”, “black” becomes “2”, and so on.



ADR Crosswalk

- Series of tables (organized by major data categories; e.g.: system data, client data, etc.)
- Columns include:
  - ID: each variable has been assigned a value in the ADR for referencing
  - ADR Variable Name (and Local System Variable Name): descriptive name
  - ADR Allowed Values (and Local System Values): list of allowed data values
  - Notes: for YOUR use to record data transformation steps, etc.

Client Level Data – Client Demographics					
ID	ADR Var Name	ADR Allowed Values	Local System Allowed Values	Local System Var Name	Notes
5	RaceID	*Multiple selections allowable (1 to 5) with "Unknown" value being mutually exclusive.*			
		1 = White			
		2 = Black or African American			
		3 = Asian			
		4 = Native Hawaiian/Pacific Islander			
		5 = Native American or Alaskan Native			
		6 = Unknown			

8

The **ID:** column identifies the variable in the *ADR Client-Level Data Report*. Each variable has been assigned a value for convenient referencing.

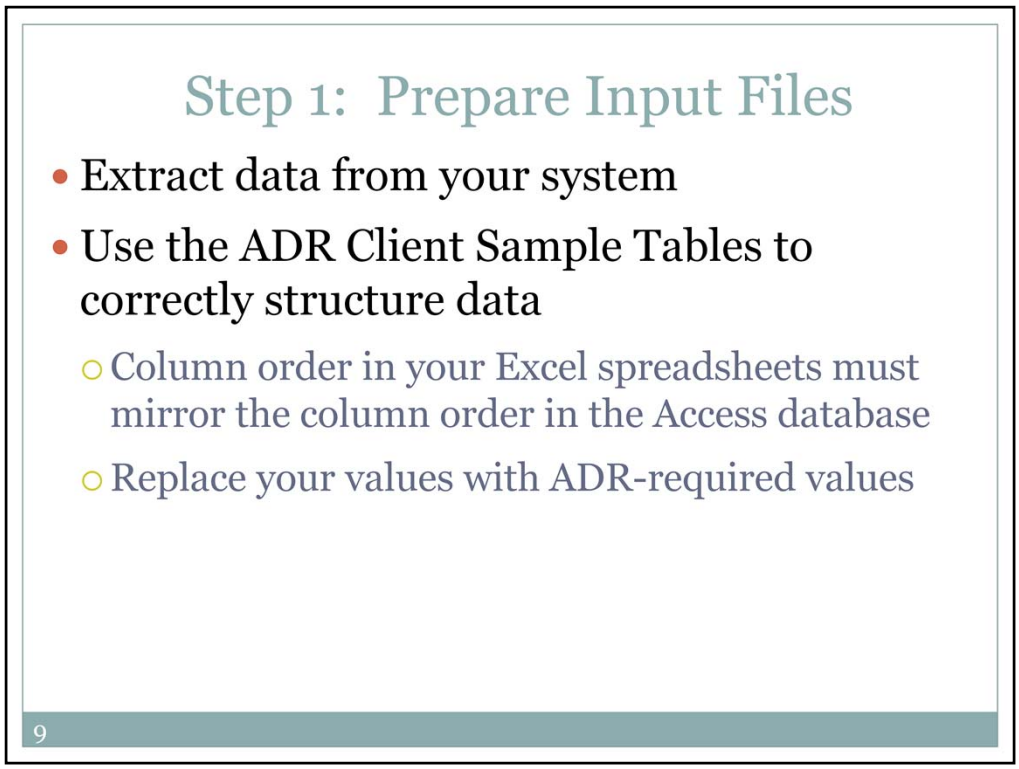
The **ADR Variable Name**: column is the descriptive name of the variable used to provide more information about what is being collected. This corresponds to the label for the variable in the *ADR Client-Level XML Schema Definitions.*

The **ADR Allowed Values**: column specifies the type or list of values allowed for the data element in the *ADR Client-Level Data.*

The **Local System Values:** column specifies the type or list of values allowed for the data element in YOUR existing dataset. Presenting this information will be useful in identifying differences between the current variable and the corresponding ADR "destination" variable.

The **Local System Variable Name:** column is the descriptive name of the variable you currently use in YOUR data.

The **Notes:** column is intended for your use to keep a record of the data transformations that you have to perform in order to provide the required ADR client-level data elements and allowed values, as shown in the previous slide.


Step 1: Prepare Input Files
Extract data from your system
Use the ADR Client Sample Tables to correctly structure data
Column order in your Excel spreadsheets must mirror the column order in the Access database
Replace your values with ADR-required values
9

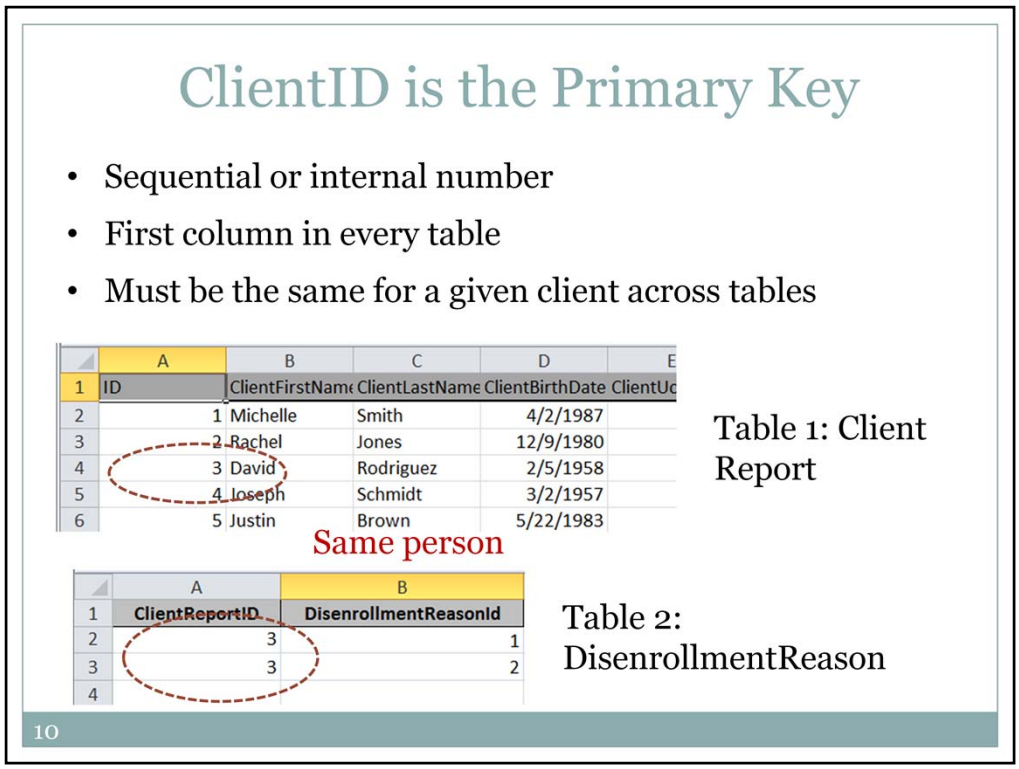
After you have completed the ADR Crosswalk, you need to extract client-level data from your system. Your data may be in multiple systems.

Then, you need to structure your data to mirror Rx-REX's Access tables. As I mentioned before, there are five tables in Rx-REX. So, you need to create five identical tables in Excel. The column order of your tables should be identical to the column order of the tables in Access. You also need to replace your values with the ADR-required values. You can do this through the replace all function or formulas in Excel. To identify the valid ADR values, you can use the data dictionary, the ADR Crosswalk, the lookup tables in Access, or the tables themselves in Access have drop down menus.

We have created a template to help you structure your data. Let's walk through it. There are five main tables in Access, so there is a tab for every table. The first table has the information you need to create the eUCI, first name, last name, and date of birth. It also contains all the data elements that just require one value per client, such as ethnicity and pregnancy status.

The next tables contain data elements for which you can provide multiples responses per client:

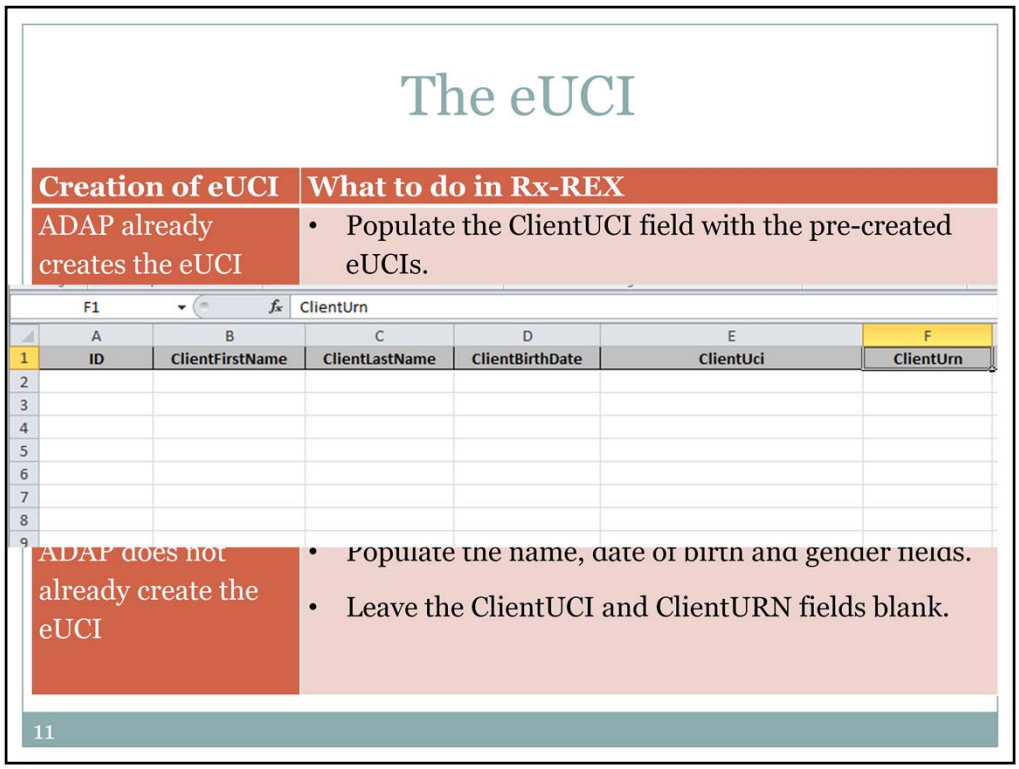
- Disenrollment reason
- Medical insurance
- Medication, which contains a row for every medication a client received in the reporting period.
- And Race



Now, I want to talk about some important data elements.

The first is the Client ID. This is the Primary Key, meaning that it is the data element that links data for a given client across the tables. It is the first column in every table.

You want to make sure that the same client has the same ID in each table. Let's look at an example. David has an ID of 3. David is the only client that has dis-enrolled from the ADAP. So, you will see the ClientReportID in the DisenrollmentReason table is 3. David has two disenrollment reason, so there are 2 rows in this table.

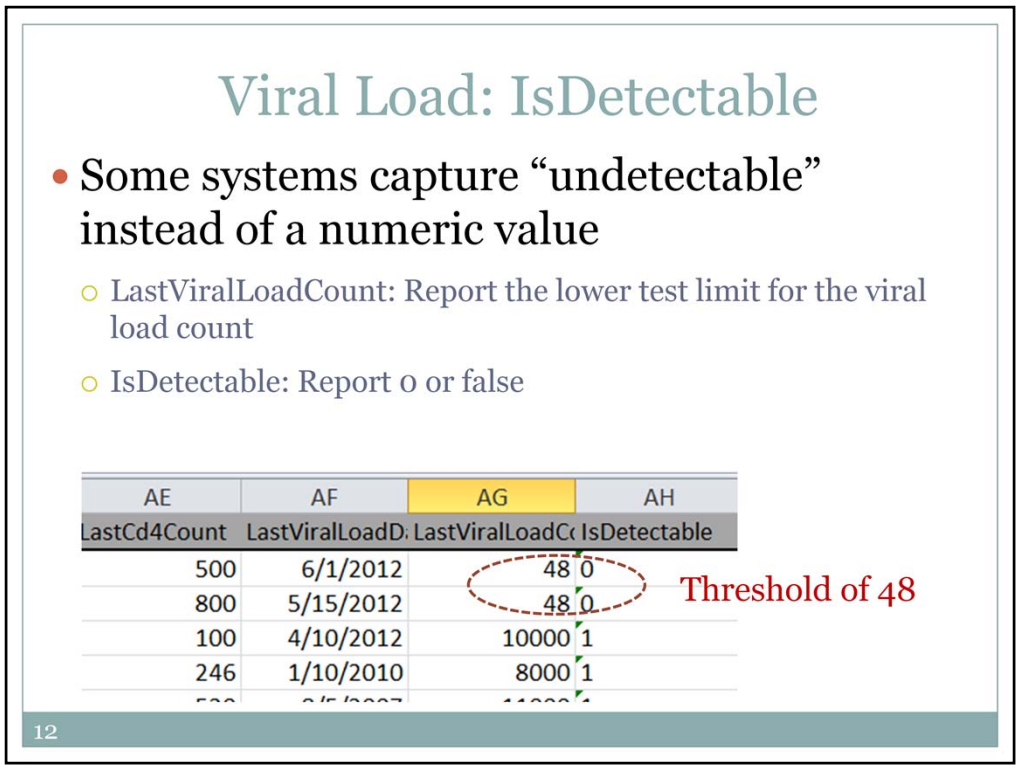


Another important data element is the eUCI. As I mentioned, there are multiple columns for client identifiers. You don't have to populate all the columns. The ones you populate depend on whether your agency already creates the eUCI or not.

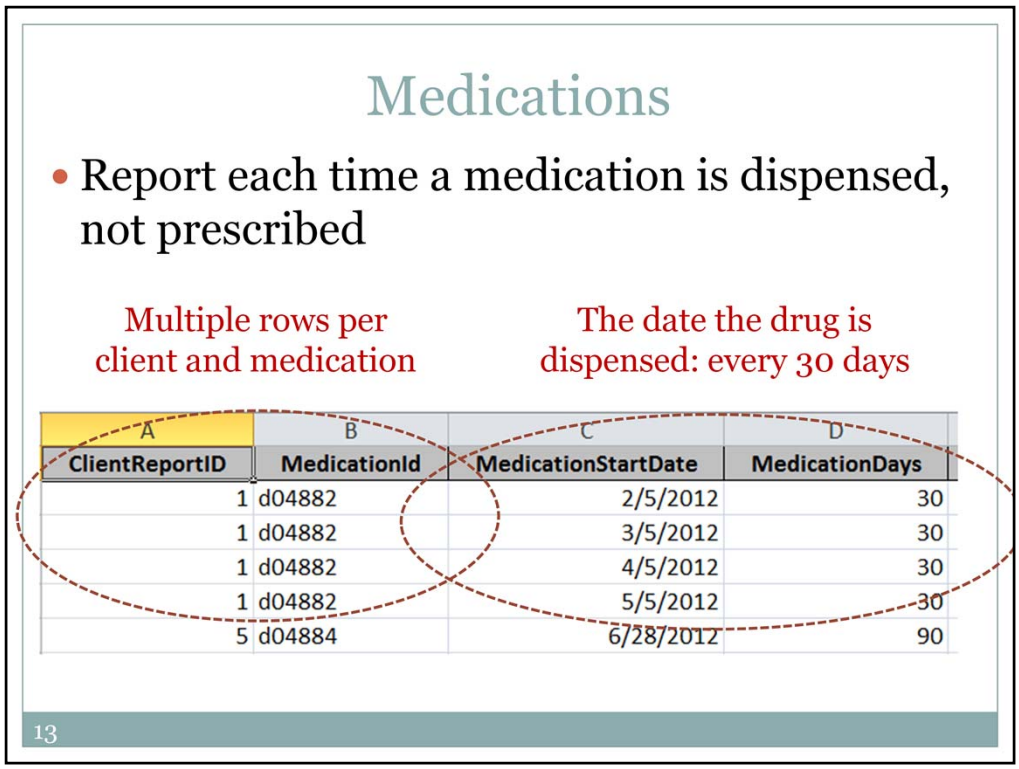
Here is a table that summarizes how to populate these columns. If you already create the eUCI using the exact same specifications and those required by the ADR, you can populate the ClientUCI column and leave the URN and name and full date of birth columns blank.

If you create the **un**encrypted UCI (or the URN), you should populate the URN columns. Rx-REX will encrypt those URNs for you.

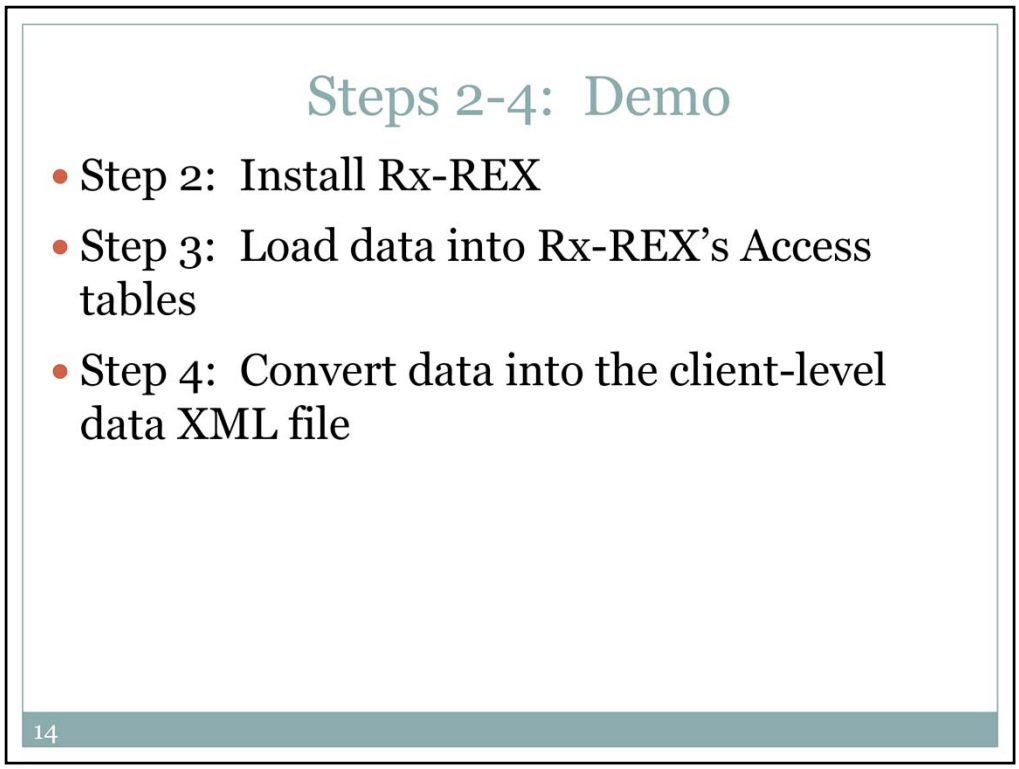
If you don't create any of these, you'll need to populate the name and date columns. We assume that the vast majority of ADAPs fall into this category.



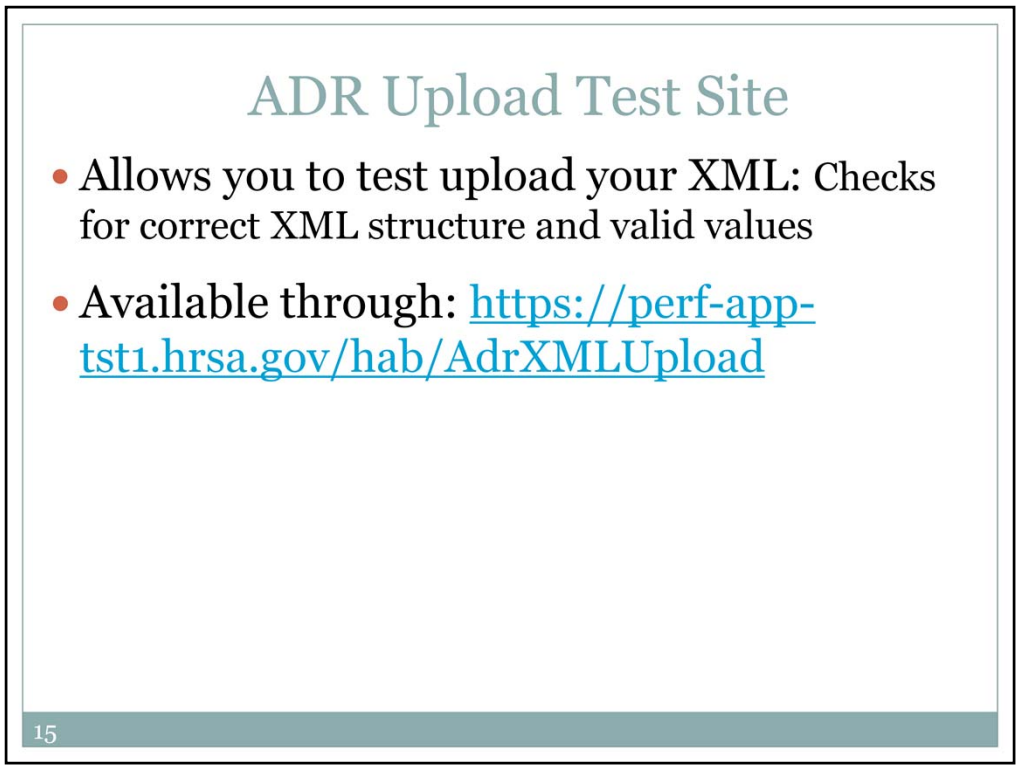
Other data elements that we have gotten questions about are related to viral load. Not all systems capture a numeric value for viral loads that are undetectable. Values under a certain threshold are just captured as "undetectable." The ADR does not accept text or carrots for the viral load value. Therefore, you should input the threshold value in the lastviralloadcount field and false or 0 in the IsDetectable field.



Finally, I want to note that in the medications tab, you report each time a drug is dispensed. So, you may have multiple rows per client and multiple rows per drug. The medication start date is the date that the drug is dispensed to the client. Medication days is the days supply of that given dispense.


Steps 2-4: Demo
Step 2: Install Rx-REX
Step 3: Load data into Rx-REX's Access tables
Step 4: Convert data into the client-level data XML file
14

Now, I'm going to hand the presentation over to Aaron, who will demo the next three steps.


ADR Upload Test Site
Allows you to test upload your XML: Checks for correct XML structure and valid values
Available through: https://perf-app-tst1.hrsa.gov/hab/AdrXMLUpload
15

Finally, let's talk briefly about the ADR Test Upload Site. SAIC created a site that allows you to test your XML file. The site checks for the correct XML structure and valid values. Any XML file created by Rx-REX should pass the test site with flying colors.

Please note that the test site does not check for data completeness and data validation issues.

You can start uploading your XML today through the following link. We'll be sending out more information to the ADAPs

Questions?

- Start with the DART Team: Data.TA@caiglobal.org
- We may connect you with the developers at SAIC